1.   (Amended) A method for processing a database query, comprising:

partially pre-aggregating records in a database according to a single grouping column to provide a result that contains at least two records having like grouping column values; and

aggregating records derived from the partial pre-aggregation to provide a result that contains records having unique grouping column values.

2.   The method as recited in claim 1, wherein the partially pre-aggregating further comprises:

maintaining a record store in memory, the record store having one record for each different grouping column value encountered in the operation;

receiving a new record;

combining the new record with a record having the same grouping column value, if such a record exists; and

adding the new record to the record store in the memory if there is no record in the record store that has the same grouping column value as the new record.

3.   The method as recited in claim 2, further comprising:

adding additional new records to the record store until the record store reaches a capacity such that it can accept no new records; and

outputting one or more records from the record store to a subsequent database operator.

4.   The method as recited in claim 3, wherein after the one or more records have been output to the subsequent database operator, the adding and outputting are repeated until there are no new records to process.

5.   The method as recited in claim 4, wherein any records remaining in the record store after there are no new records to process are output to the subsequent database operator.

6.   The method as recited in claim 3, wherein the subsequent database operator is a join.

7.   The method as recited in claim 1, further comprising estimating the costs and benefits of the partial pre-aggregation, and partially pre-aggregating the records only if the estimating indicates that the benefits are greater than the costs.

8.   The method as recited in claim 1, wherein the partially pre-aggregating includes utilizing a hashing function.

9.   The method as recited in claim 1, wherein the partial pre-aggregating creates a record store in memory, and wherein the method further comprises utilizing the record store in memory for one or more other database operators.

10.   One or more computer-readable media having computer-executable instructions that, when executed by a computer, perform the method recited in claim 1.

11.   A computer programmed to perform the method recited in claim 1.

12.   (Amended) A relational database system, comprising:

memory for storing a record store, the memory having a portion available for query processing;

a query processor coupled to the memory to process a query on the record store according to a single grouping column, the query processor being configured to partially pre-aggregate the record store to provide a result that contains at least two data records that have like grouping column values; and

the query processor being further configured to aggregate data records resulting from the partial pre-aggregation to provide an aggregation result that contains data records, no two of the data records having the same grouping column value.

13.   (Amended) The relational database system as recited in claim 12, wherein the query processor being configured to partially pre-aggregate the record store further comprises the query processor being configured to:

maintain a record store in the volatile memory, the record store having one record for each different grouping column value encountered in the partial pre-aggregation;

receive an input record from the non-volatile memory;

combine the input record with a record in the record store that has the same grouping column value, if there is such a record; and

adding the input record to the record store if there is no record in the record store that has the same grouping column value as the input record.

14.   The relational database system as recited in claim 13, wherein the query processor is further configured to:

add additional input records to the record store in memory until the record store reaches a capacity such that it can accept no more input records; and

output the records in the record store to a subsequent database operator.

15.   The relational database system as recited in claim 14, wherein the query processor is configured to:

continually add input records to the record store; and

output one or more records from the record store to a subsequent database operator when the record store reaches a capacity such that it can accept no more new records, or whenever there are no new records to process.

16.   The relational database system as recited in claim 12, wherein the query processor is further configured to perform a join on the records resulting from the partial pre-aggregation.

17.   The relational database system as recited in claim 12, wherein the query processor is further configured to create a record store as a result of the partial pre-aggregation and utilize the record store in processing of another database operator.

18.   The relational database system as recited in claim 12, further comprising:

a query optimizer configured to estimate the costs and benefits of the query processor performing a partial pre-aggregation; and

wherein the query processor performs the partial pre-aggregation only if the query optimizer indicates that the benefits of the partial pre-aggregation are greater than the costs of the partial pre-aggregation.

19.   The relational database system as recited in claim 12, wherein the query processor is further configured to utilize hashing to perform the partial pre-aggregation.

20.   The relational database system as recited in claim 12, wherein the query processor is further configured to utilize hashing and partitioning to perform the partial pre-aggregation.

21.   (Amended) A relational database computer program embodied on a computer-readable medium, comprising:

partial pre-aggregation code to partially pre-aggregate data records according to grouping column values in a single grouping column to provide a partial pre-aggregation result having two or more records having like grouping column values; and

aggregation code to aggregate data records in the partial pre-aggregation result to provide an aggregation result having records with unique grouping column values.

22.   The relational database computer program as recited in claim 21, wherein the partial pre-aggregation code is designed to:

maintain a record store in memory, the record store having one record for each different grouping column value encountered in the partial pre-aggregation;

receive a new record;

combine the new record with a record in the record store having the same grouping column value, if such a record exists;

add the new record to the record store no record in the record store that has the same grouping column value as the new record;

continuously add additional new records to the record store until the record store has reached a record store capacity;

output one or more records from the record store to a subsequent database operator when the record store has reached the record store capacity; and

output the records in the record store when there are no new records to process.

23.   The relational database computer program as recited in claim 22, further comprising database operator code that utilizes the record store for input.

24.   (Amended) A relational database computer program stored on a computer-readable medium, the relational database computer program comprising computer-executable instructions that, when executed on a computer, perform the following steps:

receiving a stream of input records;

aggregating the input records in the stream according to a single grouping column as it is received to create a record store;

joining records in the record store with other data;

outputting the records in the record store after the join;

aggregating the records output from the join; and

wherein the records output from the join include at least two records that have an identical grouping column value in the single grouping column.

25.   The relational database computer program as recited in claim 24, wherein:

the record store has a capacity that is less than the number of records in the stream of input records; and

the aggregating each input record is performed until the record store reaches capacity.

26.   The relational database computer program as recited in claim 24, further comprising computer-executable instructions that, when executed by a computer, perform the following steps:

determining if it is optimal to aggregate the input records prior to performing the join; and